Campus de l'Université de Montréal
C.P. 6128, succursale Centre ville
Montréal QC H3C 3J7 Canada

Telephone (514) 343-7575
Telecopieur (514) 343-7121

crt@crt.umontreal.ca
www.crt.umontreal.ca

CRT-2002-36

Centre de recherche sur les transports (C.R.T.)
Centre for Research on Transportation

METAHEURISTICS IN COMBINATORIAL OPTIMIZATION

by

Michel Gendreau
Jean-Yves Potvin

October 2002

103-06370/

PUBLICATION
CRT-2002-36

METAHEURISTICS IN COMBINATORIAL OPTIMIZATION*

by

Michel Gendreau[1]
Jean-Yves Potvin[1]

* This research was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) and by the Fonds pour la Formation de Chercheurs et l'Aide à la Recherche (FCAR).

[1] Département d'informatique et de recherche opérationnelle and Centre de recherche sur les transports, Université de Montréal, C.P. 6128, succursale Centre-ville, Montréal, Canada H3C 3J7 <michelg@crt.umontreal.ca, potvin@crt.umontreal.ca>

Dépôt légal – Bibliothèque nationale du Québec
Bibliothèque nationale du Canada, 2002

Résumé. L'émergence d'approches métaheuristiques pour résoudre des problèmes complexes d'optimisation combinatoire est l'une des découvertes les plus marquantes des deux dernières décennies en recherche opérationnelle. Cet article présente les résultats les plus récents dans ce domaine et identifie des problématiques et tendances communes. Des exemples d'applications pour différents problèmes de génération de tournées de véhicules sont également présentés.

Mots clés: métaheuristiques, optimisation par colonies de fourmis, algorithmes génétiques, GRASP, recherche dispersée, recuit simulé, recherche tabou, recherche à voisinage variable, optimisation combinatoire, tournées de véhicules.

Abstract. The emergence of metaheuristics for solving difficult combinatorial optimization problems is one of the most notable achievements of the last two decades in operations research. This paper provides an account of the most recent developments in the field and identifies some common issues and trends. Examples of applications are also reported for vehicle routing and scheduling problems.

Keywords: metaheuristics, ant colony optimization, genetic algorithms, GRASP, scatter search, simulated annealing, tabu search, variable neighborhood search, combinatorial optimization, vehicle routing.

1 Introduction

This paper provides an overview of metaheuristics and their application to combinatorial optimization problems. Although a large number of references are found at the end, this paper is not intended to be a complete survey. Rather, it provides a glimpse at the rapid evolution of metaheuristic concepts, their convergence towards a unified framework, and the richness of potential applications (using vehicle routing problems for illustrative purposes). For the sake of brevity, only a subset of metaheuristics is examined, which represents a good sample of the different search paradigms proposed in the field, namely: the greedy randomized adaptive search procedure, simulated annealing, tabu search, variable neighborhood search, ant colony optimization, evolutionary algorithms and scatter search. For a broader view on metaheuristics, one may refer to the following books, monographies and edited volumes [2, 26, 63, 67, 77, 85, 92, 100, 102, 112, 114, 127]. Also, an overwhelming number of conferences (and conference proceedings) are devoted to this subject.

In the early years, specialized heuristics were typically developed to solve complex combinatorial optimization problems. With the emergence of more general solution schemes (Glover coined in 1986 the term "metaheuristics" for such methods [58]), the picture drastically changed. Now, the challenge is to adapt a metaheuristic to a particular problem or problem class, which usually requires much less work than developing a specialized heuristic from scratch. Furthermore, a good metaheuristic implementation is likely to provide near-optimal solutions in reasonable computation times. With the advent of increasingly powerful computers and parallel platforms, metaheuristics have even been successfully applied to real-time problems with stringent response time requirements (see, for example, [52, 55]).

In the following section, we first introduce a number of well-known metaheuristics and emphasize their most recent developments. From this discussion, common issues and trends emerge, leading to a unifying view of metaheuristics which is presented in Section 3. For illustrative purposes, Section 4 then explains how the various metaheuristic search schemes can be adapted to vehicle routing and scheduling, a prominent class of problems in combinatorial optimization. Finally, concluding remarks follow in Section 5.

2 Metaheuristics

In this section, metaheuristics are divided into two categories: ***single-solution*** metaheuristics where a single solution (and search trajectory) is considered at a time and ***population*** metaheuristics where a multiplicity of solutions evolve concurrently. Within each category, it would also be possible to distinguish between primarily constructive metaheuristics, where a solution is built from scratch (through the introduction of new elements at each iteration) and improvement

metaheuristics which iteratively modify a solution. In the two following subsections, the constructive metaheuristics are represented by the greedy randomized adaptive search procedure and ant colony optimization, respectively. The others can be considered as primarily improvement metaheuristics.

2.1 Single-Solution Metaheuristics

2.1.1 GRASP

GRASP (Greedy Randomized Adaptive Search Procedure) is basically a multi-start process, where each restart consists of applying a randomized greedy construction heuristic to create a new solution, which is then improved through a local descent [43, 44]. This is repeated for a given number of restarts and the best overall solution is returned at the end. At each step of the construction heuristic, the elements not yet incorporated into the partial solution are evaluated with a greedy function, and the best ones are kept in a so-called "restricted candidate list" (RCL). One element is then randomly chosen from this list and incorporated into the solution. Through randomization, the best current element is not necessarily chosen, thus leading to a diversity of solutions. Recent surveys and tutorials on GRASP may be found in [45, 83, 103, 113].

One shortcoming of GRASP is that each restart is independent of the previous ones. Recent developments are thus aimed at devising mechanisms to allow the search to learn from previously obtained solutions. One example is the reactive GRASP [106, 107], where the size of the RCL is dynamically adjusted, depending on the quality of recently generated solutions. Another example is the use of memories to guide the search. In [46], a pool of elite solutions is maintained to bias the probability distribution associated with the elements in the RCL. Intensification or diversification can be obtained by either rewarding or penalizing elements that are often found in the pool of elite solutions. Such a pool can also be used to implement path relinking [65], by generating a search trajectory between a randomly chosen elite solution and the current local optimum [83].

Other recent developments propose hybrids where the construction step is followed by more sophisticated local search procedures, in particular variable neighborhood search [91]. Parallel implementations are also reported, mostly based on the distribution of the restarts over multiple processors.

2.1.2 Simulated Annealing (SA)

SA is a randomized local search procedure where a modification to the current solution leading to an increase in solution cost can be accepted with some probability. This algorithm is motivated from an analogy with the physical annealing process used to find low-energy states of solids [20, 79]. In a combi-

natorial optimization context, a solution corresponds to a state of the physical system and the solution cost to the energy of the system. At each iteration, the current solution is modified by randomly selecting a move from a particular class of transformations (which defines a neighborhood of solutions). If the new solution provides an improvement, it is automatically accepted and becomes the new current solution. Otherwise, the new solution is accepted according to the Metropolis criterion, where the probability of acceptance is related to the magnitude of the cost increase and a parameter called the "temperature". Basically, a move is more likely to be accepted if the temperature is high and the cost increase is low. The temperature parameter is progressively lowered, according to some predefined cooling schedule, and a certain number of iterations are performed at each temperature level. When the temperature is sufficiently low, only improving moves are accepted and the method stops in a local optimum. As opposed to most metaheuristics, this method asymptotically converges to a global optimum (assuming an infinite number of iterations). Finite-time implementations, however, do not provide such a guarantee.

Developments around this basic search scheme have focused on the followings [1, 3, 75]:

- convergence results for generalized simulated annealing procedures, based on more general forms of acceptance rule than the Metropolis one.
- deterministic variants with threshold accepting, where a transition is accepted if it does not increase the cost by more than some predefined value; this value is progressively reduced as the algorithm unfolds [41].
- different forms of static and dynamic cooling schedules aimed at increasing the speed of convergence, without compromising solution quality.
- parallel annealing where various parts of the algorithm are distributed over a number of parallel processors [6, 126].
- hybrids with other metaheuristics, in particular genetic algorithms and neural networks (to control self-organization).

2.1.3 Tabu Search (TS)

TS is basically a deterministic local search strategy where, at each iteration, the best solution in the neighborhood of the current solution is selected as the new current solution, even if it leads to an increase in solution cost. As opposed to a pure local descent, the method will thus escape from a local optimum. A short-term memory, known as the tabu list, stores recently visited solutions (or attributes of recently visited solutions) to avoid short-term cycling. Typically, the search stops after a fixed number of iterations or a maximum number of consecutive iterations without any improvement to the incumbent (best known) solution. The principles of the method originates from the work of Glover

in [58]. Detailed descriptions of TS, including recent developments, may be found in [49, 50, 59, 60, 62, 63, 64, 66].

Starting from the simple search scheme described above, a number of developments and refinements have been proposed over the years. These include:

- Introduction of intensification and diversification mechanisms. These are typically implemented via different forms of long-term memories [63].
  - frequency memories are used to record how often certain solution attributes are found in previously visited solutions. Neighborhood solutions which contain elements with high frequency counts can then be penalized to allow the search to visit other regions of the search space. This mechanism provides a form of "continuous" diversification by introducing a bias in the evaluation of neighborhood solutions at each iteration.
  - adaptive memories provide a means to both diversify and intensify the search [115]. Such memories contain a pool of previously generated elite solutions, which can be used to restart the search. This is usually done by taking different fragments of elite solutions and by combining them to generate a new starting solution, similarly to many population metaheuristics (see below). Intensification or diversification is obtained depending if the fragments are taken from solutions that lie in a common region of the search space or not. Adaptive memories provide a generic paradigm for guiding local search and can be coupled with different types of metaheuristics, as discussed in Section 3.
  - path relinking [63, 65] generates new solutions by exploring trajectories between elite solutions. Starting from one of these solutions, it generates a path in the neighborhood space leading to another solution, called the guiding solution. This can be done by selecting modifications that introduce attributes found in the guiding solution. This mechanism can be used to diversify or intensify the search, depending on the path generation mechanism and the choice of the initiating and guiding solutions.
  - In strategic oscillation [63], an oscillation boundary (usually, a feasibility boundary) is defined. Then, the search is allowed to go for a specified depth beyond the boundary before turning around. When the boundary is crossed again from the opposite direction, the search goes beyond it for a specified depth before turning around again. By repeating this procedure an oscillatory search pattern is produced. It is possible to vary the amplitude of the oscillation to explore a particular region of the search space. For example, tight oscillations favor a more thorough search around the boundary.

- The reactive tabu search [7] provides a mechanism for dynamically adjusting the search parameters, based on the search history. In particular, the size of tabu list is automatically increased when some configurations are repeated too often to avoid short-term cycles (and conversely).
- the probabilistic TS [63] is aimed at introducing randomization into the search method. This can be achieved by associating probabilities with moves leading to neighborhood solutions, based on their evaluation. As a complete neighborhood evaluation is often too computationally expensive, randomization can also be introduced in the design of a candidate list strategy, where only a subset of neighborhood solutions is considered.
- The latest implementations of TS are often quite cumbersome, due to the inclusion of many additional components requiring a lot of parameters. Unified tabu search [22] represents a recent trend aimed at producing simpler, more flexible TS code. Its main characteristics are simple, easily adaptable, neighborhood structures, coupled with a dynamic adjustment of (a few) parameters. Another example of this new trend is the granular TS [125].

2.1.4 Variable Neighborhood Search (VNS)

VNS is based on the idea of systematically changing the neighborhood structure within a local search heuristic, rather than using a single neighborhood structure [94]. Given a set of pre-selected neighborhood structures (which are often nested), a solution is randomly generated in the first neighborhood of the current solution, from which a local descent is performed. If the local optimum obtained is not better than the incumbent, then the procedure is repeated with the next neighborhood. The search restarts from the first neighborhood when either a solution which is better than the incumbent has been found or every neighborhood structure has been explored. A popular variant is the deterministic Variable Neighborhood Descent (VND) where the best neighbor of the current solution is considered instead of a random one. Also, no local descent is performed with this neighbor. Rather, it automatically becomes the new current solution if an improvement is obtained, and the search is then restarted from the first neighborhood. Otherwise, the next neighborhood is considered. The search stops when all neighborhood structures have been considered and no improvement is possible. At this point, the solution is a local optimum for all neighborhood structures. Surveys and tutorials about VNS and VND may be found in [69, 70, 71, 72, 73].

One interesting development is the Variable Neighborhood Decomposition Search (VNDS) where the kth neighborhood consists of fixing all but k attributes (or variables) in the current solution. The local descent is then performed on the k free variables only [74]. Another variant is the skewed VNS where the evaluation of the current local optimum is biased with the distance from the

incumbent [68]. The idea is to favor the exploration of distant solutions from the incumbent to diversify the search. Recently, hybrids with tabu search have been reported [70], where either TS is used within VNS or conversely. In the first case, the local descent in VNS is replaced by a tabu search while, in the second case, different neighborhood structures are exploited by TS. Hybrids with a constructive metaheuristic like GRASP are also reported, where VNS or VND is used for the improvement step. Finally, several ways to parallelize VNS have been proposed in the literature [70]. One natural approach is to generate many random neighbors from the current solution (either in the same neighborhood or in different neighborhoods) and to assign each neighbor to a different processor to perform the local descent [28].

2.2 Population Metaheuristics

This class of metaheuristics explicitly works with a population of solutions (rather than a single solution). In such a case, different solutions are combined, either implicitly or explicitly, to create new solutions.

2.2.1 Ant Colony Optimization (ACO)

This metaheuristic is motivated from the way real ants find shortest paths leading from the nest to food sources, by laying a chemical compound, known as pheromone, on the ground. Basically, a number of artificial ants construct solutions in a randomized and greedy way. Each ant chooses the next element to be incorporated into its current partial solution on the basis of some heuristic evaluation and the amount of pheromone associated with that element. The latter represents the memory of the system, and is related to the presence of that element in previously constructed solutions. The stochastic element in the ACO metaheuristic allows the ants to build a variety of different solutions. Basically, a probability distribution is defined over all elements which may be incorporated into the current partial solution, with a bias in favor of the best elements. That is, an element with a good heuristic evaluation and a high level of pheromone is more likely to be selected.

Starting from the original ant system framework described in [34] for solving the Traveling Salesman Problem, a number of refinements have since been integrated to lead to the ACO. These refinements provide mechanisms to either intensify or diversify the search. For example, more pheromone may be associated with elements found in the incumbent, to favor a more intense search around that solution. Conversely, the pheromone levels may be reduced for some elements, to force the construction of diversified solutions. For a precise description of ACO and a historical perspective, see [35, 36, 37, 39, 40, 90, 121].

Recent developments include the followings:

- the Max-Min Ant System [118, 119], where the pheromone levels are bounded;
- the Hybrid Ant System (HAS) [48], where the pheromone trails are used to guide a local search heuristic rather than a construction heuristic;
- the use of multiple ant colonies which interact by exchanging information about good solutions, in particular pheromone trails [47, 93];
- the exploitation of more sophisticated greedy heuristics to construct solutions [88];
- the coupling of ACO with local search to improve the solutions constructed by the artificial ants;
- the design of parallel implementations; ACO, like other population-based metaheuristics, lends itself quite naturally to such implementations, by devising subcolonies evolving on different processors. This is quite similar to the island model of genetic algorithms (see below).

2.2.2 Evolutionary Algorithms (EAs)

Evolutionary algorithms represent a large class of problem-solving methodologies, with genetic algorithms (GAs) being one of the most widely known [77]. These algorithms are motivated by the way species evolve and adapt to their environment, based on the Darwinian principle of natural selection. Under this paradigm, a population of solutions (often encoded as a bit or integer string, referred to as a "chromosome") evolves from one generation to the next through the application of operators that mimic those found in nature, namely, selection of the fittest, crossover and mutation. Through the selection process, which is probabilistically biased towards the best elements in the population, only the best solutions are allowed to become "parents" and to generate "offspring". The mating process, called crossover, then takes two selected parent solutions and combine their most desirable features to create one or two offspring solutions. This is repeated until a new population of offspring solutions is created. Before replacing the old population, each member of the new population is subjected (with a small probability) to small random perturbations via the mutation operator. Starting from a randomly or heuristically generated initial population, this renewal cycle is repeated for a number of iterations, and the best solution found is returned at the end. Details about this metaheuristic, and its application to combinatorial optimization problems, may be found in [8, 67, 92, 98, 104].

Genetic algorithms have been successfully applied to many combinatorial optimization problems, although this success has often been achieved by departing from the classical scheme depicted above. In particular, the encoding of solutions into "chromosomes" is either completely ignored (by working directly on the solutions) or specifically designed for specialized crossover and mutation

operators. The distinctive feature of GAs remains the exploitation of a population of solutions and the creation of new solutions through the recombination of good attributes of two parent solutions. Many single-solution metaheuristics now integrates this feature (e.g., via adaptive memories).

One of the main improvements comes from the observation that, although the population improves on average over the generations, it often fails to generate near-optimal solutions. Thus, some form of intensification is needed, and this is usually achieved in modern hybrid GAs by considering powerful local search operators as a form of mutation. That is, rather than introducing small random perturbations into the offspring solution, a local search is applied to improve the solution until a local optimum is reached [84, 95, 96, 97, 108]. This way, selection and crossover sample the search space with solutions that are then processed by the local search. Fined-grained and coarse-grained parallel implementations, are also gaining popularity [18]. In the former case, each solution is implemented on a different processor and is allowed to mate only with solutions that are not too distant in the underlying connection topology. This approach may be implemented in a totally asynchronous way, with each solution evolving independently of the others. Restricting the pool of mating candidates to a subset of the population has beneficial effects of its own, by allowing good attributes to diffuse more slowly into the population, thus preventing premature convergence to grossly suboptimal solutions. In coarse-grained implementations, subpopulations of solutions evolve in parallel until migration of the best individuals from one subpopulation to another takes place to "refresh" the genetic pool. This so-called "island model" however requires some form of synchronization.

2.2.3 Scatter Search (SS)

Scatter search combines a set of solutions, called the reference set, to create new solutions. Convex and non convex combinations of a variable number of solutions are performed to create candidates inside and outside the convex regions spanned by the reference set. This mechanism for generating new solutions is similar to the one found in genetic algorithms, except that many "parent" solutions (not necessarily two) are combined to create "offspring". The candidate solutions are then heuristically improved or repaired (i.e., projected in the space of feasible solutions) and the best ones, according to their quality and diversity, are added to the reference set, if they are good enough. Starting from some initial reference set, usually created through heuristic means, this procedure is repeated until the reference set does not change anymore. At this point, it is possible to restart from another initial reference set.

The basic ideas of SS originate from [56, 57]. A precise description of this metaheuristic may be found in [65, 82].

3 Common Issues and Trends

Although the metaheuristics described in section 2 may look quite different in their algorithmic principles, a closer analysis of the most recent implementations unveil common issues and trends. That is, a unification or convergence phenomenon is observed over the years. In fact, a GRASP exploiting memories between its restarts, and where each solution is further improved with a local search method, is not much different from a probabilistic TS with restarts provided through adaptive memories. These points of convergence are identified and briefly discussed in the followings.

Breadth-first versus depth-first. One common issue is certainly the search for a good trade-off between a wide but shallow exploration of the solution space versus a more focused and intense exploration in certain regions. This is sometimes referred to as diversification versus intensification or exploration versus exploitation (with, however, slightly different interpretations).

Dynamic parameter adjustments. Reactive implementations, where the parameters are automatically adjusted during the search, are now widely used. This approach provides robustness to the metaheuristic, by avoiding a manual fine-tuning for each problem under study.

Memories. Apart from simulated annealing, which offers a somewhat more rigid framework, the use of memories to guide the search is now a fundamental ingredient of metaheuristics. This is a way of providing learning mechanisms based on the search history. Memories have always been associated with TS, although other mechanisms, which can be considered as memories, are at the core of other metaheuristics as well. This is the case, for example, with pheromone trails in ACO. The current population of solutions in GAs, which is the result of an evolution process over many generations, can also be seen as a form of memory. Finally, recent implementations of various metaheuristics, like VNS and GRASP, incorporate memories. Adaptive memories, in particular, are now widely used. In fact, they represent a new search paradigm of their own, called Adaptive Memory Programming (AMP) [63, 123], since they provide starting points and guide other metaheuristics (a kind of meta-metaheuristic, in a sense). When AMP is coupled with a local descent, the method is referred to as Adaptive Descent. This simple search scheme has been shown to be very efficient in [52]. Basically, the adaptive memory samples the search space with new starting points, which are then improved by a descent method until a local optimum is obtained. In [89], a framework called Iterated Local Search (ILS) generalizes these ideas by providing different perturbation mechanisms to sample the solution space in the search of good local optima.

Randomization. Randomization is inherently present in most metaheuristics. We can think about the Metropolis acceptance rule of SA, the probabilistic parent selection scheme and the mutation operator of GAs, the randomized greedy heuristics of GRASP and ACO. Even TS, which was originally designed as a

deterministic search method, often exploits randomization to select a solution in the current neighborhood (thus, alleviating the risks of cycling). In general, randomization favors the creation of a diversity of solutions, by allowing a better exploration of the search space.

Recombination. Population heuristics provide a natural framework for combining good features of two or more solutions to create new solutions. This recombination process has been integrated into single-solution metaheuristics through various means, in particular, via adaptive memories and path relinking. The latter approach is somewhat intriguing, as it does not directly merge the initiating and guiding solutions to produce a new solution, but rather generates a search trajectory between the two solutions.

Large-Scale Neighborhoods. Large-scale neighborhood structures allow a more thorough search of the solution space. These include, for example, ejection chains, cyclic transfer and sequential fan and look ahead strategies [5, 61]. These large neighborhoods are typically coupled with filtering techniques aimed at focusing the search on the most interesting solutions, as a complete evaluation of the entire neighborhood would be extremely costly. These sophisticated structures are often required to obtain optimal or quasi-optimal solutions for complex combinatorial problems. At the other end of the spectrum, many researchers are now looking for more straightforward metaheuristic implementations with relatively simple neighborhood structures and only a few parameters [21]. This approach brings simplicity, flexibility and robustness to the algorithm. Furthermore, high quality solutions can still be obtained in a matter of a few seconds (rather than minutes or hours).

Hybrids. Hybrids are aimed at exploiting the strengths of different methods by either designing a new metaheuristic that combines various algorithmic principles of the original methods or by developing multi-agent architectures in which the individual agents run pure methods and communicate among themselves [120]. Recent experiments on parallel platforms have confirmed the effectiveness of the latter multi-agent scheme [27, 86].

Parallel implementations. Among the various implementation reported in the literature, parallel exploration strategies where several search threads run concurrently seem to be very promising [29, 30, 32]. When applied to single-solution metaheuristics, this organization also blurs the distinction with population metaheuristics. Important issues in this context relate to the communication scheme (if any) among the search threads. One answer consists in coupling parallel search with adaptive memories. Under this scheme, threads exchange information by sending their good solutions to a central, common adaptive memory, which is then used to restart the threads. Other schemes partition the search space in a fixed or adaptive fashion.

Based on these observations, a unifying view of metaheuristics emerges. This is illustrated in Figure 1, using a relatively small number of algorithmic components. These are:

- *Construction.* This is used by all metaheuristics for creating the first initial solution(s). GRASP and ACO also use this component for creating other solutions (as they are iterative construction metaheuristics).
- *Recombination.* This component generates new solutions from the current ones through a recombination process. Most heuristics bypass this component, except GA and SS.
- *Random Modification.* This is used to modify the current solution(s) through random perturbations.
- *Improvement.* This is used to explicitly improve the current solution(s), for example, by selecting the best solution in the neighborhood, by applying a local descent or by projecting the solution in the feasible region.
- *Memory Update.* This component updates either standard memories (TS), pheromone trails (ACO), populations (GA) or reference sets (SS).
- *Parameter/Neighborhood Update.* This component adjusts the parameter values or modifies the neighborhood structures (as in VNS).

In the figure, it is possible to find the path associated with each metaheuristic, by following the labeled links. This path corresponds to the basic implementation of each metaheuristic. For example, GRASP goes only through the *Construction* and *Improvement* components, since it was originally defined as a multi-start process where new solutions are iteratively constructed and improved with a local descent. Recent developments now integrate *Memory Update*, by using memories to avoid independent restarts, and *Parameter/Neighborhood Update*, by dynamically adjusting the size of the restricted candidate list. Similarly, GA goes through *Construction* to create an initial population, *Recombination* to generate offspring, *Random Modification* (mutation) and *Memory Update* to perform generation replacement. Recent developments now integrate the *Improvement* component, by applying local search to individual solutions. It is thus interesting to note that the most recent metaheuristic developments are aimed at integrating more components from Figure 1, thus leading to the convergence phenomenon observed over the years.

4 Application Example

Metaheuristics have been successfully applied to a wide range of combinatorial optimization problems. It would thus be a long and ambitious task to provide an exhaustive account (for such an effort up to 1996, see [101]). Successful implementations are reported for routing, scheduling, production planning, network design, quadratic assignment and many others. For illustrative purposes, this section will thus focus on a particular class of problems, namely vehicle

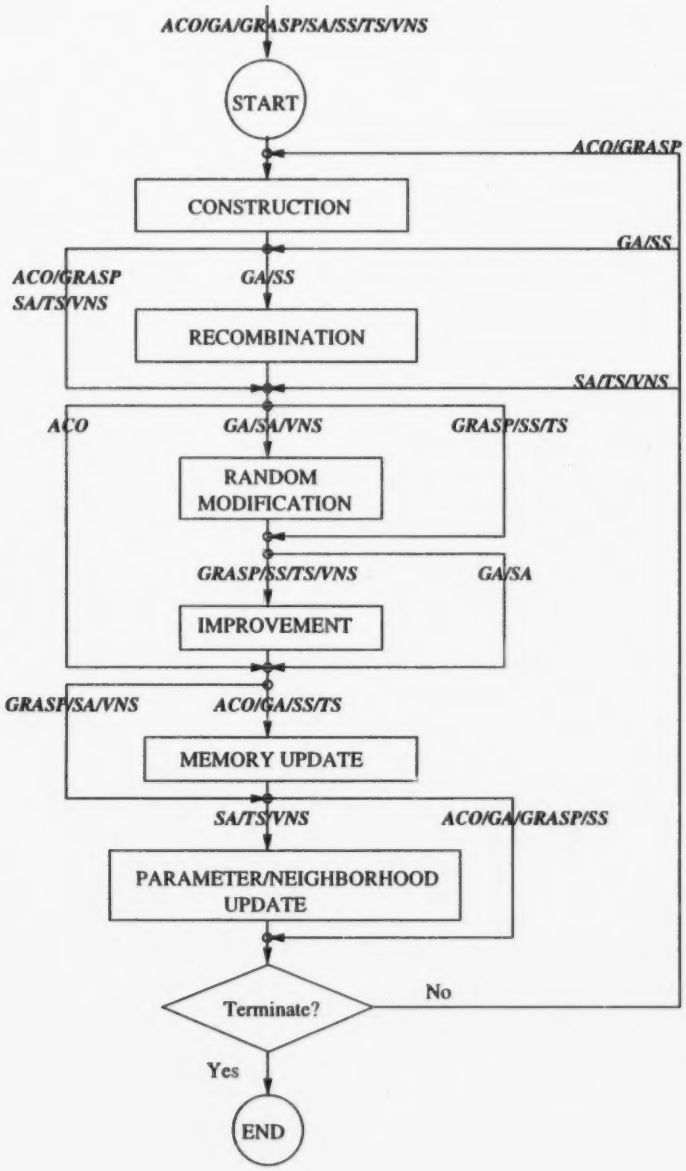

Figure 1: A Unifying View of Metaheuristics

routing and scheduling problems, as they exhibit an impressive record of successful metaheuristic implementations and are used to model many real-world applications (e.g., distribution of goods, emergency and repair services, etc.). For each metaheuristic introduced in Section 2, Table 1 provides references for the most widely known subclasses of problems, namely:

- the Traveling Salesman Problem (TSP) where a single route of minimum distance that visits every node exactly once is constructed.
- the capacitated Vehicle Routing Problem (VRP) where a capacity constraint (in terms of weight, volume or maximum distance) limits the number of serviced nodes on each route.
- the VRP with time windows (VRPTW), where a time interval for service is associated with each node.
- the VRP with backhauls (VRPB), where each node represents either a pick-up or a delivery.
- the Pick-up and Delivery Problem (PDP) where each customer consists of a pick-up and a delivery node, with a precedence constraint between the two. That is, some commodity must first be taken at the pick-up location before being brought to the delivery location. This is to be opposed to the previous subclasses of problems where a customer is represented by a single node.

Whenever possible, surveys are referred to in Table 1 to avoid too many references. Since each survey itself refers to a long list of papers, the number of metaheuristic implementations in this field is clearly impressive. Furthermore, we have restricted ourselves to only a subset of vehicle routing problems, leaving apart variants like multi-depot vehicle routing, periodic vehicle routing, inventory-routing and many others.

For vehicle routing and scheduling, as well as for any other application, the metaheuristic under study must be adapted to the problem at hand. In the following, we briefly explain how each metaheuristic can be tailored to these problems.

i) ACO was originally designed for the TSP. At each step, the ant chooses the next node to be visited using a greedy evaluation function based on the distance to be traveled from the current node, as well as the amount of pheromone on the corresponding edge. Since the choice of the next node to be visited is based on a probability distribution (biased towards the nodes with better evaluations), each ant applies a probabilistic nearest neighbor heuristic to construct a tour. When side constraints are present, they are handled by considering only nodes that can be feasibly inserted in the current route, and by creating a new route when there is no such node.

ii) GAs implementations often use the integer string representation to encode a solution, where each integer stands for a particular node. Thus, a string really represents a permutation of nodes (i.e., a route). Specialized crossover and mutation operators, including local search, are then designed to generate new offspring permutations from parent ones [104]. In the case of multiple-route solutions, the representation must also include "separators" between routes. Different approaches are proposed in the literature to handle side constraints. These include:

- the *decoder* approach, where the evolutionary algorithm searches the space of permutations and calls a greedy heuristic to create a feasible solution from the current permutation (by sequentially inserting the nodes into the routes, based on the permutation order).
- the *penalty* approach where the evaluation of infeasible solutions is penalized. This way, an infeasible solution is less likely to be selected for mating and will tend to disappear from the population.
- the *repair* approach, where an infeasible solution is transformed into a feasible one through carefully designed repair operators.

iii) GRASP applies classical randomized greedy insertion heuristics to construct a solution, followed by local search. Side constraints can be easily handled during the construction process by considering only feasible insertion places for the nodes (and by creating a new route when there is none).

iv) For SA, TS and VNS a central issue is the design of appropriate neighborhood structures. These are typically obtained by *moving* or *exchanging* nodes or edges. For example, a node can be moved at another position in the same route or in another route. Alternatively, a node can exchange its position with some other node in the same route or in another route. More sophisticated operators extend these types of modifications by combining them to create ejection chains [61, 109, 111] or consider sequences of many nodes [99, 122]. Infeasible solutions are typically penalized, to look less attractive when the neighborhood of the current solution is evaluated. The decoder and repair approaches, presented in the context of EAs, can also be used here to deal with side constraints.

5 Conclusion

Metaheuristics are powerful algorithmic approaches which have been applied with great success to many difficult combinatorial optimization problems. One nice feature of metaheuristics is that they can quite easily handle the complicating constraints found in real-life applications. However, this does not mean that metaheuristics can be applied blindly to any new problem. Significant

Problems	ACO	GA	GRASP	SA	SS	TS	VNS
TSP	[36, 38] [40, 88] [117]	[4, 78] [92, 104]		[78, 81]	[110]	[78, 110]	[17, 69] [70, 94]
VRP	[15, 16] [36, 40] [54]	[53, 54]	[45, 76]	[53, 54]		[24, 53] [54]	
VRPTW	[14, 47]	[12, 14]	[45, 80]	[14]		[13, 14]	[9, 11] [14, 25] [116]
VRPB		[105]	[19, 45]			[42]	[31]
PDP		[10, 87]				[23, 33] [51, 124]	

Table 1: Vehicle Routing Applications

knowledge about the problem is required to develop a successful metaheuristic implementation. In particular, the generic search scheme provided by metaheuristics must be carefully adapted to the specific characteristics of the problem (e.g. by choosing an appropriate search space and effective neighborhood structures). As previously observed, many well-known metaheuristics seem to converge towards a unifying framework made of a few algorithmic components. In this unifying process, new opportunities for combining the strengths (and alleviating the weaknesses) of these methods will emerge, thus leading to even more powerful search models for difficult combinatorial optimization problems.

Acknowledgments. Financial support for this work was provided by the Natural Sciences and Engineering Research Council of Canada (NSERC) and by the Québec Fonds pour la Formation de Chercheurs et l'Aide à la Recherche (FCAR). This support is gratefully acknowledged.

References

[1] Aarts E.H.L., J.H.M. Korst, P.J.M. van Laarhoven, "Simulated Annealing", in *Local Search in Combinatorial Optimization*, E.H.L. Aarts and J.K. Lenstra (eds), 91–136, Wiley, 1997.

[2] Aarts E.H.L., J.K. Lenstra (eds), Local Search in Combinatorial Optimization, Wiley, 1997.

[3] Aarts E.H.L, H.M.M. Ten Eikelder, "Simulated Annealing", in *Handbook of Applied Optimization*, P.M. Pardalos and M.G.C. Resende (eds), Oxford University Press, New York, 209–220, 2002.

[4] Aarts E.H.L., M.G.A. Verhoeven, "Genetic Local Search for the Traveling Salesman Problem", in *Handbook of Evoluationary Computation*, T. Bäck, D.B. Fogel, Z. Michalewicz (eds), G9.5, Oxford University Press, 1997.

[5] Ahuja R.K., O. Ergun, J.B. Orlin, A.P. Punnen, "A Survey of Very Large-Scale Neighborhood Search Techniques", Working Paper, forthcoming in *Discrete Applied Mathematics*, 2002.

[6] Azencott R. (ed.), Simulated Annealing: Parallelization Techniques, Wiley, 1992.

[7] Battiti R., G. Tecchiolli, "The Reactive Tabu Search", *ORSA Journal on Computing 6*, 126–140, 1994.

[8] Beasley J.E., "Population Heuristics", in *Handbook of Applied Optimization*, P.M. Pardalos and M.G.C. Resende (eds), Oxford University Press, New York, 138-156, 2002.

[9] Bent R., P. Van Hentenryck, "A Two-Stage Hybrid Local Search for the Vehicle Routing Problem with Time Windows", Technical Report CS-01-06, Dept. of Computer Science, Brown University, 2001.

[10] Benyahia I., J.-Y. Potvin, "Decision Support for Vehicle Dispatching using Genetic Programming", *IEEE Transactions on Systems, Man and Cybernetics 28*, 306–314, 1998.

[11] Bräysy O., "Local Search and Variable Neighborhood Search Algorithms for the Vehicle Routing Problem with Time Windows", Doctoral Dissertation, University of Vaasa, Finland, 2001.

[12] Bräysy O., M. Gendreau, "Genetic Algorithms for the Vehicle Routing Problem with Time Windows", SINTEF Report STF42 A01022, Oslo, Norway, 2001.

[13] Bräysy O., M. Gendreau, "Tabu Search Heuristics for the Vehicle Routing Problem with Time Windows", SINTEF Report STF42 A01021, Oslo, Norway, 2001.

[14] Bräysy O., M. Gendreau, "Metaheuristics for the Vehicle Routing Problem with Time Windows", SINTEF Report STF42 A01025, Oslo, Norway, 2001.

[15] Bullnheimer B., R.F. Hartl, C. Strauss, "Applying the Ant System to the Vehicle Routing Problem", in *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*, S. Voss, S. Martello, I.H., C. Roucairol (eds), 285–296, Kluwer, 1999.

[16] Bullnheimer B., R.F. Hartl, C. Strauss, "An Improved Ant System for the Vehicle Routing Problem", *Annals of Operations Research 89*, 319–328, 1999.

[17] Burke E.K., P. Cowling, R. Keuthen, "Effective Local and Guided Variable Neighborhood Search Methods for the Asymmetric Traveling Salesman Problem", *Lecture Notes in Computer Science 2037*, 203–212, 2001.

[18] Cantú-Paz E., Efficient and Accurate Parallel Genetic Algorithms, Kluwer, 2000.

[19] Carreto C., B. Baker, "A GRASP Interactive Approach to the Vehicle Routing Problem with Backhauls", in *Essays and Surveys in Metaheuristics*, C.C. Ribeiro, P. Hansen (eds), 185–200, Kluwer, 2001.

[20] Cerny V., "Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm", *Journal of Optimization Theory and Applications 45*, 41–51, 1985.

[21] Cordeau J.-F., M. Gendreau, G. Laporte, J.-Y. Potvin, F. Semet, "A Guide to Vehicle Routing Heuristics", *Journal of the Operational Research Society 53*, 512–522, 2002.

[22] Cordeau J.-F., G. Laporte, A. Mercier, "A Unified Tabu Search Heuristic for Vehicle Routing Problems with Time Windows", *Journal of the Operational Research Society 52*, 928–936, 2001.

[23] Cordeau J.-F., G. Laporte, "A Tabu Search Heuristic for the Static Multi-Vehicle Dial-a-Ride Problem", Technical Report G-2002-18, Groupe d'études et de recherche en analyse des décisions, Montreal, 2002.

[24] Cordeau J.-F., G. Laporte, "Tabu Search Heuristics for the Vehicle Routing Problem", Technical Report G-2002-15, Groupe d'études et de recherche en analyse des décisions, Montréal, 2002.

[25] Cordone R., R.W. Calvo, "A Heuristic for the Vehicle Routing Problem with Time Windows", *Journal of Heuristics 7*, 107–129, 2001.

[26] Corne D., M. Dorigo, F. Glover (eds), New Ideas in Optimization, McGraw-Hill, 1999.

[27] Crainic T.G., M. Gendreau, "Towards an Evolutionary Method - Cooperative Multi-Thread Parallel Tabu Search Heuristic Hybrid", in *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*, S.Voss, S. Martello, I.H. Osman and C. Roucairol (eds), Kluwer, 331–344, 1999.

[28] Crainic T.G., M. Gendreau, P. Hansen, N. Hoeb, N. Mladenović, "Parallel Variable Neighborhood Search for the p-Median", in *Proceedings of MIC'2001*, Porto, 595–599, 2001.

[29] Crainic T.G., M. Toulouse, "Parallel Strategies for Meta-Heuristics", Technical Report CRT-2001-06, Centre de recherche sur les transports, Montréal, 2001 (forthcoming in Handbook of Metaheuristics, F. Glover and G. Kochenberger (eds), Kluwer).

[30] Crainic T.G., M. Toulouse, M. Gendreau, "Towards a Taxonomy of Parallel Tabu Search Algorithms", *INFORMS Journal on Computing 9*, 61–72, 1997.

[31] Crispim J., J. Brandao, "Reactive Tabu Search and Variable Neighborhood Descent Applied to the Vehicle Routing Problem with Backhauls", in *Proceedings of MIC'2001*, 931-636, Porto, 2001.

[32] Cung V.-D., S.L. Martins, C.C. Ribeiro, C. Roucairol, "Strategies for the Parallel Implementation of Metaheuristics", in *Essays and Surveys in Metaheuristics*, C.C. Ribeiro and P. Hansen (eds), Kluwer, 263–308, 2002.

[33] Desaulniers G., J. Desrosiers, A. Erdmann, M.M. Solomon, F. Soumis, "VRP with Pickup and Delivery", in *The Vehicle Routing Problem*, P. Toth and D.Vigo (eds), SIAM 225–242, 2001.

[34] Dorigo M., Optimization, Learning, and Natural Algorithms, Ph.D. Thesis, Politecnico di Milano, 1992.

[35] Dorigo M., G. Di Caro, "The Ant Colony Optimization Meta-Heuristic", in *New Ideas in Optimization*, D. Corne, M. Dorigo and F. Glover (eds), McGraw-Hill, 11–32, 1999.

[36] Dorigo M., G. Di Caro, L.M. Gambardella, "Ant Algorithms for Discrete Optimization", *Artificial Life 5*, 137–172, 1999.

[37] Dorigo M., L.M. Gambardella, "Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem", *IEEE Transactions on Evolutionary Computation 1*, 53–66, 1997.

[38] Dorigo M., L.M. Gambardella, "Ant Colonies for the Traveling Salesman Problem", *BioSystems 43*, 73–81, 1997.

[39] Dorigo M., V. Maniezzo, A. Colorni, "The Ant System: Optimization by a Colony of Cooperating Agents", *IEEE Transactions on Systems, Man, and Cybernetics B26*, 29–41, 1996.

[40] Dorigo M., T. Stützle, "The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances", Technical Report IRIDIA/2000-32, Université Libre de Bruxelles, Belgium, 2000 (forthcoming in *Handbook of Metaheuristics*, F. Glover and G. Kochenberger (eds), Kluwer).

[41] Dueck G., T. Scheuer, "Threshold Accepting: A General Purpose Optimization Algorithm", *Journal of Computational Physics 90*, 161–175, 1990.

[42] Duhamel C., J.-Y. Potvin, J.-M. Rousseau, "A Tabu Search Heuristic for the Vehicle Routing Problem with Backhauls and Time Windows", *Transportation Science 31*, 49–59, 1997.

[43] Feo T.A., M.G.C. Resende, "A Probabilistic Heuristic for a Computationally Difficult Set Covering Problem", *Operations Research Letters 8*, 67–71, 1989.

[44] Feo T.A., M.G.C. Resende, "Greedy Randomized Adaptive Search Procedures", *Journal of Global Optimization 6*, 109–133, 1995.

[45] Festa P., M.C.G. Resende, "GRASP: An Annotated Bibliography", in *Essays and Surveys in Metaheuristics*, C.C. Ribeiro and P. Hansen (eds), Kluwer, 325–368, 2002.

[46] Fleurent C., F. Glover, "Improved Constructive Multistart Strategies for the Quadratic Assignment Problem using Adaptive Memory", *INFORMS Journal on Computing 11*, 198–204, 1999.

[47] Gambardella L.-M., E.D. Taillard, G. Agazzi, "MACS-VRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows", in *New Ideas in Optimization*, D. Corne, M. Dorigo and F. Glover (eds), McGraw-Hill, 63–76, 1999.

[48] Gambardella L.-M., E.D. Taillard, M. Dorigo, "Ant Colonies for the Quadratic Assignment Problem", *Journal of the Operational Research Society 50*, 167–176, 1999.

[49] Gendreau M., "An Introduction to Tabu Search", forthcoming in *Handbook of Metaheuristics*, F. Glover and G. Kochenberger (eds), Kluwer, 2003.

[50] Gendreau M., "Recent Advances in Tabu Search", in *Essays and Surveys in Metaheuristics*, C.C. Ribeiro and P. Hansen (eds), 369–378, Kluwer, 2002.

[51] Gendreau M., F. Guertin, J.-Y. Potvin, R. Séguin, "Neighborhood Search Heuristics for a Dynamic Vehicle Dispatching Problem with Pick-ups and Deliveries", Technical Report CRT-98-10, Centre de recherche sur les transports, Université de Montréal, 1998.

[52] Gendreau M., F. Guertin, J.-Y. Potvin, E.D. Taillard, "Parallel Tabu Search for Real-Time Vehicle Routing and Dispatching", *Transportation Science 33*, 381–390, 1999.

[53] Gendreau M., G. Laporte, J.-Y. Potvin, "Vehicle Routing: Modern Heuristics" in *Local Search in Combinatorial Optimization*, E.H.L. Aarts and J.K. Lenstra (eds), 311–336, Wiley, 1997.

[54] Gendreau M., G. Laporte, J.-Y. Potvin, "Metaheuristics for the Capacitated VRP" in *The Vehicle Routing Problem*, P. Toth and D. Vigo (eds), SIAM, 129–154, 2002.

[55] Gendreau M., G. Laporte, F. Semet, "A Dynamic Model and Parallel Tabu Search Heuristic for Real-Time Ambulance Relocation", *Parallel Computing 27*, 1641–1653, 2001.

[56] Glover F., "Parametric Combinations of Local Job Shop Rules", Chapter IV, ONR Research Memorandum no. 117, GSIA, Carnegie-Mellon University, Pittsburgh, U.S.A., 1963.

[57] Glover F., "Heuristics for Integer Programming using Surrogate Constraints", *Decision Sciences 8*, 156–166, 1977.

[58] Glover F., "Future Paths for Integer Programming and Links to Artificial Intelligence", *Computers & Operations Research 13*, 533–549, 1986.

[59] Glover F., "Tabu Search - Part I", *ORSA Journal on Computing 1*, 190–206, 1989.

[60] Glover F., "Tabu Search - Part II", *ORSA Journal on Computing 2*, 4–32, 1990.

[61] Glover F., "Ejection Chains, Reference Structures and Alternating Path Methods for the Traveling Salesman Problem", *Discrete Applied Mathematics 65*, 223–253, 1996.

[62] Glover F., "Tabu Search and Adaptive Memory Programming - Advances, Applications and Challenges", in *Advances in Metaheuristics, Optimization and Stochastic Modeling Technologies*, R.S. Barr, R.V. Helgason and J.L. Kennington (eds), Kluwer, Boston, 1–75, 1997.

[63] Glover F., M. Laguna, Tabu Search, Kluwer, 1997.

[64] Glover F., M. Laguna, "Tabu Search", in *Handbook of Applied Optimization*, P.M. Pardalos and M.G.C. Resende (eds), Oxford University Press, New York, 194–208, 2002.

[65] Glover F., M. Laguna, R. Marti, "Fundamentals of Scatter Search and Path Relinking", *Control and Cybernetics 39*, 653–684, 2000.

[66] Glover F., M. Laguna, E.D. Taillard, D. de Werra (eds), Tabu Search, *Annals of Operations 41*, 1993.

[67] Goldberg D.E., Genetic Algorithms in Search, Optimization & Machine Learning, Addison-Wesley, Reading, 1989.

[68] Hansen P., B. Jaumard, N. Mladenović, A. Parreira, "Variable Neighborhood Search for Weighted Maximum Satisfiability Problem", Les Cahiers du GERAD, G-2000-62, Montréal, 2000.

[69] Hansen P., N. Mladenović, "An Introduction to Variable Neighborhood Search", in *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*, S. Voss, S. Martello, I.H. Osman and C. Roucairol (eds), Kluwer, 433–458, 1999.

[70] Hansen P., N. Mladenović, "Variable Neighborhood Search, a Chapter of Handbook of Metaheuristics", Les Cahiers du GERAD, G-2001-41, Montréal, 2001 (forthcoming in *Handbook of Metaheuristics*, F. Glover and G. Kochenberger (eds), Kluwer).

[71] Hansen P., N. Mladenović, "Variable Neighborhood Search: Principles and Applications", *European Journal of Operational Research 130*, 449–467, 2001.

[72] Hansen P., N. Mladenović, "Developments of Variable Neighborhood Search", in *Essays and Surveys in Metaheuristics*, C.C. Ribeiro and P. Hansen (eds), Kluwer, 415–440, 2002.

[73] Hansen P., N. Mladenović, "Variable Neighborhood Search", in *Handbook of Applied Optimization*, P.M. Pardalos and M.G.C. Resende (eds), Oxford University Press, New York, 221–234, 2002.

[74] Hansen P., N. Mladenović, D. Perez-Brito, "Variable Neighborhood Decomposition Search", *Journal of Heuristics 7*, 335–350, 2001.

[75] Henderson D., S.H. Jacobson, A.W. Johnson, "The Theory and Practice of Simulated Annealing", forthcoming in *Handbook of Metaheuristics*, F. Glover and G. Kochenberger (eds), Kluwer, 2003.

[76] Hjorring C.A., "The Vehicle Routing Problem and Local Search Metaheuristics", Ph.D. Thesis, University of Auckland, 1995.

[77] Holland J.H., Adaptation in Natural and Artificial Systems, The University of Michigan Press, 1975.

[78] Johnson D.S., McGeoch L.A., "The Traveling Salesman Problem: A Case Study", in *Local Search in Combinatorial Optimization*, E.H.L. Aarts and J.K. Lenstra (eds), 215–310, Wiley, 1997.

[79] Kirkpatrick S., C.D. Gelatt Jr., M.P. Vecchi, "Optimization by Simulated Annealing", *Science 220*, 671–680, 1983.

[80] Kontoravdis G., J.F. Bard, "A GRASP for the Vehicle Routing Problem with Time Windows", *ORSA Journal on Computing 7*, 10–23, 1995.

[81] Koulamas C., S.R. Antony, R. Jaen, "A Survey of Simulated Annealing Applications to Operations Research Problems", *Omega 22*, 41–56, 1994.

[82] Laguna M., "Scatter Search", in *Handbook of Applied Optimization*, P.M. Pardalos and M.G.C. Resende (eds), Oxford University Press, New York, 183–193, 2002.

[83] Laguna M., R. Martí, "GRASP and path relinking for 2-layer straight line crossing minimization", *INFORMS Journal on Computing 11*, 44–52, 1999.

[84] Land M., "Evolutionary Algorithms with Local Search for Combinatorial Optimization", Ph.D. Dissertation, University of California, San Diego, 1998.

[85] Laporte G., I.H. Osman (eds), Metaheuristics in Combinatorial Optimization, *Annals of Operations Research 63*, 1996.

[86] LeBouthillier A., T.G. Crainic, R. Keller, "Parallel Cooperative Evolutionary Algorithm for Vehicle Routing Problems with Time Windows", in *Proceedings of Odysseus 2000*, 78–81, Chania, 2000.

[87] Leclerc F., Potvin J.-Y., "Genetic Algorithms for Vehicle Dispatching", *International Transactions in Operational Research 4*, 391–400, 1997.

[88] Lelouarn F.-X., Gendreau M., Potvin J.-Y., "GENI Ants for the Traveling Salesman Problem", Technical Report CRT-2001-40, Centre de recherche sur les transports, Montréal, 2001.

[89] Lorenço H.R., O. Martin, T. Stützle, "Iterated Local Search", forthcoming in *Handbook of Metaheuristics*, F. Glover and G. Kochenberger (eds), Kluwer, 2003.

[90] Maniezzo V., A. Carbonaro, "Ant Colony Optimization: An Overview", in *Essays and Surveys in Metaheuristics*, C.C. Ribeiro and P. Hansen (eds), 469–492, Kluwer, 2002.

[91] Martins S.L., M.G.C. Resende, C.C. Ribeiro and P. Pardalos, "A Parallel GRASP for the Steiner Tree Problem in Graphs using a Hybrid Local Search Strategy", *Journal of Global Optimization 17*, 267–283, 2000.

[92] Michalewicz Z., Genetic Algorithms + Data Structures = Evolution Programs, Third Edition, Springer, Berlin, 1996.

[93] Middendorf M., F. Reischle, H. Schmeck, "Multi Colony Ant Algorithms", *Journal of Heuristics 8*, 305–320, 2002.

[94] Mladenović N., P. Hansen, "Variable Neighborhood Search", *Computers & Operations Research 24*, 1097–1100, 1997.

[95] Moscato P., "Memetic Algorithms: A short introduction", in *New Ideas in Optimization*, D. Corne, M. Dorigo and F. Glover (eds), McGraw-Hill, 219–234, 1999.

[96] Moscato P., "Memetic Algorithms", in *Handbook of Applied Optimization*, P.M. Pardalos and M.G.C. Resende (eds), Oxford University Press, New York, 157–167, 2002.

[97] Mühlenbein H., "How Genetic Algorithms Really Work: Mutation and Hill-Climbing", in *Parallel Problem Solving from Nature 2*, R. Männer and B. Manderick (eds), North-Holland, Amsterdam, 15–26, 1992.

[98] Mühlenbein H., "Genetic Algorithms" in *Local Search in Combinatorial Optimization*, E.H.L. Aarts and J.K. Lenstra (eds), Wiley, Chichester, 137–171, 1997.

[99] Osman I.H., "Metastrategy Simulated Annealing and Tabu Search Algorithms for the Vehicle Routing", *Annals of Operations Research 41*, 421–452, 1993.

[100] Osman I.H., J.P. Kelly (eds), Meta-Heuristics: Theory & Applications, Kluwer, 1996.

[101] Osman I.H., G. Laporte, "Metaheuristics: A Bibliography", *Annals of Operations Research 63*, 513–623, 1996.

[102] Pardalos P.M., M.G.C. Resende (eds), Handbook of Applied Optimization, Chapter 3.6 on Metaheuristics, Oxford University Press, New York, 123-234, 2002.

[103] Pitsoulis L.S., M.G.C. Resende, "Greedy Randomized Adaptive Search Procedures", in *Handbook of Applied Optimization*, P.M. Pardalos and M.G.C. Resende (eds), Oxford University Press, New York, 168–182, 2002.

[104] Potvin J.-Y., "Genetic Algorithms for the Traveling Salesman Problem", *Annals of Operations Research 63*, 339–370, 1996.

[105] Potvin J.-Y., C. Duhamel, F. Guertin, "A Genetic Algorithm for Vehicle Routing with Backhauling", *Applied Intelligence 6*, 345–355, 1996.

[106] Prais M., C.C. Ribeiro, "Reactive GRASP: An Application to a Matrix Decomposition Problem", *INFORMS Journal on Computing 12*, 164–176, 2000.

[107] Prais M., C.C. Ribeiro, "Parameter Variation in GRASP procedures", *Investigación Operativa 9*, 1–20, 2000.

[108] Preux P., E-G. Talbi, "Towards hybrid evolutionary algorithms", *International Transactions in Operational Research 6*, 557–570, 1999.

[109] Rego C., "Node Ejection Chains for the Vehicle Routing Problem: Sequential and Parallel Algorithms", *Parallel Computing 27*, 201–222, 2001.

[110] Rego C., F. Glover, "Local Search and Metaheuristics for the Traveling Salesman Problem", Technical Report HCES-07-01, Hearin Center for Enterprise Science, The University of Mississipi, 2001.

[111] Rego C., C. Roucairol, "A Parallel Tabu Search Algorithm Using Ejection Chains for the Vehicle Routing Problem", in *Metaheuristics: Theory and Applications*, I.H. Osman and J.P. Kelly (eds), Kluwer, Boston, 661–675, 1996.

[112] Reeves C. (ed), Modern Heuristic Techniques for Combinatorial Problems, Blackwell, Oxford, 1993.

[113] Resende M.G.C., C.C. Ribeiro, "Greedy Randomized Adaptive Search Procedures", forthcoming in *Handbook of Metaheuristics*, F. Glover and G. Kochenberger (eds), Kluwer, 2003.

[114] Ribeiro C.C., P. Hansen (eds), Essays and Surveys in Metaheuristics, Kluwer, 2002.

[115] Rochat Y., E.D. Taillard, "Probabilistic Diversification and Intensification in Local Search for Vehicle Routing", *Journal of Heuristics 1*, 147–167, 1995.

[116] Rousseau L.-M., M. Gendreau, G. Pesant, "Using Constraint-Based Operators with Variable Neighborhood Search to Solve the Vehicle Routing Problem with Time Windows" *Journal of Heuristics 8*, 43–58, 2002.

[117] Stützle T., M. Dorigo, "ACO Algorithms for the Traveling Salesman Problem", in *Evolutionary Algorithms in Engineering and Computer Science*, K. Miettinen, M. Makela, P. Neittaanmaki and J. Periaux (eds), Wiley, 1999.

[118] Stützle T., H.H. Hoos, "Improvements on the Ant System: Introducing the MAX-MIN Ant System", in *Artificial Neural Networks and Genetic Algorithms*, G.D. Smith, N.C. Steele and R.F. Albrecht (eds), 245–249, Springer Verlag, 1998.

[119] Stützle T., H.H. Hoos, "MAX-MIN Ant System", *Future Generation Computer Systems Journal 16*, 889–914, 2000.

[120] Talbi E.G., "A Taxonomy of Hybrid Metaheuristics", Technical Report AS-183, LIFL, Université des Sciences et Technologies de Lille, France, 1998.

[121] Taillard E.D., "Ant Systems", in *Handbook of Applied Optimization*, P.M. Pardalos and M.G.C. Resende (eds), Oxford University Press, New York, 130–137, 2002.

[122] Taillard E.D., Badeau P., Gendreau M., Guertin F., Potvin J.-Y., "A Tabu Search Heuristic for the Vehicle Routing Problem with Soft Time Windows", *Transportation Science 31*, 170–186, 1997.

[123] Taillard E.D., L.-M. Gambardella, M. Gendreau, J.-Y. Potvin, "Adaptive Memory Programming: A Unified View of Metaheuristics", *European Journal of Operational Research 135*, 1–16, 2001.

[124] Toth P., D. Vigo, "Heuristic Algorithms for the Handicapped Persons Transportation Problem", *Transportation Science 31*, 60–71, 1997.

[125] Toth P., D. Vigo, "The Granular Tabu Search (and its application to the vehicle routing problem)", Technical Report OR/98/9, D.E.I.S., Universita di Bologna, 1998 (forthcoming in *INFORMS Journal on Computing*).

[126] Verhoeven M.G.A., E.H.L. Aarts, "Parallel Local Search Techniques", *Journal of Heuristics 1*, 43–65, 1996.

[127] Voss S., S. Martello, I.H. Osman and C. Roucairol (eds), Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization, Kluwer, 1999.